

Perancangan Editor Model Data Konseptual dan Model Data Fisik

Suhadi Lili¹, Aldy Sefan Rezanaldy², Siti Rochimah³

^{1,2,3} Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo Surabaya, 60111, Indonesia

¹suhadi.lili@gmail.com, ²aldy.sefan@gmail.com, ³siti@its-sby.edu

Abstrak

Desain model basis data merupakan sebuah fase penting dalam pengembangan sebuah Aplikasi Sistem Informasi. Model data konseptual merupakan sebuah model pendekatan secara konsep sebelum dikonversi menjadi model data fisik yang telah berkaitan dengan tujuan basis data tertentu. Editor merupakan sebuah tool yang digunakan untuk mempermudah melakukan desain basis data dengan pendekatan model data konseptual dan sebagai jembatan penghubung antara model data konseptual dan model data fisik. Editor dapat melakukan konversi antara model data konseptual dan model data fisik. Editor yang direncanakan ini menerapkan round trip engineering, yaitu konversi bolak-balik antara data model konseptual dan data model fisik. Konversi dilakukan secara otomatis dan realtime. Perancangan Editor model data ini memaksimalkan fungsional sistem round trip engineering, dengan tujuan untuk memberikan kemudahan bagi pengguna dan perancang basis data merancang basis data dengan memanfaatkan dan memaksimalkan fungsional tersebut.

Kata kunci : Editor, Round trip engineering, Desain model basis data, Model data konseptual, Model data fisik

1. Pendahuluan

Model data konseptual merupakan cara untuk menyatakan basis data dalam bentuk yang mudah untuk dipahami dan diverifikasi kebenarannya. Model data konseptual ini biasanya berbentuk *Entity Relationship Diagram* (ERD) [2]. Model data konseptual ini terlepas dari *Database Management System* (DBMS) dan *platform* perangkat keras yang digunakan untuk mengimplementasikan basis data tersebut. Perancangan sistem informasi biasanya memulai memodelkan basis data dengan model data konseptual ini.

Model data fisik merupakan cara untuk menyatakan struktur basis data yang lebih definitif untuk salah satu target basis data tertentu, misalnya: MySQL, Microsoft SQL Server, Oracle, dan sebagainya. Model data fisik memiliki detail yang cukup untuk membentuk struktur basis data pada salah satu target tersebut secara eksak.

Konversi yang dilakukan adalah bolak-balik, yaitu dengan *Round Trip Engineering* merupakan sebuah proses konversi antara model data konseptual dan model data fisik. Dari pendekatan pengguna kepada pendekatan basis data, atau sebaliknya. Pada tools yang ada, konversi ini dilakukan secara manual dengan memberikan perintah “Generate”, sehingga mulailah dilakukan konversi dari model data konseptual ke model data fisik atau sebaliknya. Pada perancangan editor basis data ini, konversi dilakukan secara realtime, tidak memerlukan perintah “Generate” untuk melakukan konversi.

Dalam makalah ini dibahas tentang perancangan editor basis data yang menerapkan konversi bolak-balik secara otomatis, dengan memanfaatkan dan memaksimalkan fungsionalitas dari konversi bolak-balik tersebut (*Round Trip Engineering*).

2. Analisis Kebutuhan Editor Model Data

Sebagaimana Editor Data model yang lain, Editor Data Model ini memiliki kebutuhan kesamaan kebutuhan dan spesifikasi dengan yang lainnya. Diantara tujuan dari pemodelan adalah:

1. Melakukan testing pada entitas fisik sebelum dibangun,
2. Sebagai alat komunikasi dengan customer,
3. Sebagai alat alternatif visualisasi informasi, dan
4. Mengurangi kompleksitas.

Tujuan inilah yang harus tetap dipertahankan dalam merancang aplikasi editor model data. Dengan tambahan fungsional konversi bolak-balik dengan *round trip engineering* tanpa mengurangi tujuan semula dari perancangan model data.

Kebutuhan fungsional sebuah editor adalah kemampuan untuk melakukan:

1. penambahan obyek diagram,

2. manipulasi obyek diagram, dan
3. penghapusan obyek diagram.

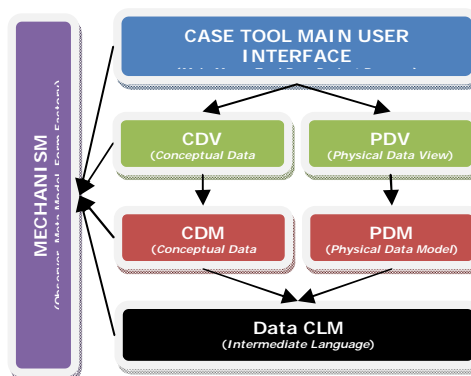
Ini merupakan kebutuhan utama dari seluruh aplikasi Editor pemodelan. Selain itu pemodelan harus memiliki sebuah standard pemodelan yg digunakan. Pada model data konseptual dan model data fisik yang paling sesuai dengan pemodelan ini adalah Object Modeling Technique (OMT) dengan penulisan Entity Relationship Diagram (ERD) dengan style “Crow’s Leg”.

3. Perancangan Sistem

Arsitektur sistem yang telah mendukung *round trip engineering* atau konversi bolak-balik secara otomatis, terdiri dari beberapa domain yang dijelaskan pada Gambar 1.

- a. *Mechanism*, menangani pembaharuan informasi dari obyek-obyek model data dan *form factory*.
- b. *Case Tool Main user Interface*, domain ini menangani tampilan antar muka utama dari Editor tool atau CASE Tool, seperti *form* menu utama dan *project browser*.
- c. *Conceptual Data View (CDV)*, domain ini sebagai penampilan diagram (diagram viewing) data model konseptual. CDV juga sebagai tempat melakukan edit dan manipulasi obyek. Di dalamnya bisa ditambahkan obyek pada diagram, dilakukan manipulasi pada obyek, dan penghapusan obyek. Obyek-obyek diagram yang ada pada domain ini adalah obyek-obyek model data konseptual.
- d. *Physical Data View (PDV)*, sebagai penampilan diagram (diagram viewing) data model physical, sebagai tempat melakukan edit dan manipulasi obyek. Di dalamnya bisa ditambahkan obyek pada diagram, dilakukan manipulasi pada obyek, dan penghapusan obyek. Obyek-obyek diagram yang ada pada domain ini adalah obyek-obyek model data fisik.
- e. *Conceptual Data Model (CDM)*, sebagai representasi dari obyek-obyek yang ada pada data model konseptual. Diantaranya, *Entity*, *Association*, *Data Item*, *Relationship*, *Inheritance*, *Association Link*, dan obyek-obyek lain yang ada pada data model konseptual.
- f. *Physical Data Model (PDM)*, sebagai representasi dari obyek-obyek yang ada pada data model fisik. Diantaranya, *Table*, *Column*, *Reference*, dan obyek-obyek lain yang ada pada model data fisik.
- g. *Data Common Language Model (Data CLM)*, merupakan sebuah domain yg merupakan media untuk melakukan konversi bolak-balik. *Data Common Language (CLM)* merupakan *super set* dari data model konseptual dan data model fisik. Istilah lainnya

adalah bahasa perantara dari data model konseptual dan data model fisik. Ketika ada perubahan pada CDV yang merubah CDM, atau pada PDV yang merubah PDM, maka akan diteruskan dan diinterpretasikan ke Data CLM dan di-*observe* kembali ke CDM dan PDM.



Gambar 1. Arsitektur Editor Model Data

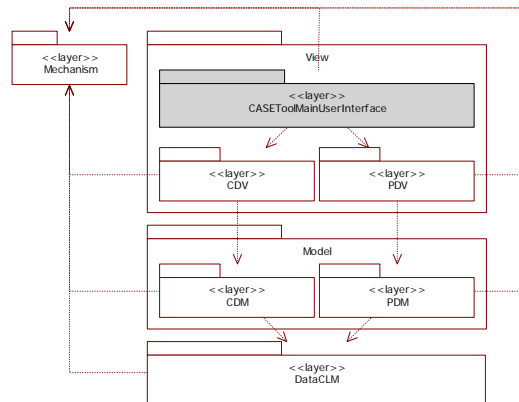
Arsitektur Model Data, memiliki tujuh domain utama yang terbagi dalam empat layer (seperti terlihat pada Gambar 2), yaitu:

- 1) *Layer View*, yaitu domain *Case Tool Main User Interface*, *CDV*, dan *PDV*
- 2) *Layer Model*, yaitu domain *CDM* dan *PDM*
- 3) *Layer Layer Data CLM*, domain *Data CLM*
- 4) *Layer Mechanism*, domain *Mechanism*

User berhubungan langsung dengan *Layer View*. *User* dapat melakukan show diagram, menambahkan obyek pada diagram canvas, memanipulasinya, menghapusnya, dan melakukan segala interaksi dengan Editor pada tampilan antar muka Editor. Seluruh interaksi *User* pada *Layer View* yang berkaitan dengan obyek model akan diteruskan pada *Layer Model*. Dari *CDV* ke *CDM*, atau dari *PDV* ke *PDM*.

Kemudian, Dari *Layer Model* akan diteruskan ke *Layer Data CLM* dengan merepresentasikan obyek-obyek pada *Layer Model* menjadi *super set* yang meliputi kedua model, yaitu *CDM* dan *PDM*. Pada *Layer Data CLM* inilah dilakukan *round trip engineering*, yaitu yang diistilahkan dengan melakukan konversi secara bolak-balik, yaitu dengan memisahkan dari *super set* yang ada pada *Data CLM* menjadi *subset* yang dikembalikan ke *Layer Model* untuk kemudian di tampilkan pada *CDV* dan *PDV* pada *Layer View*.

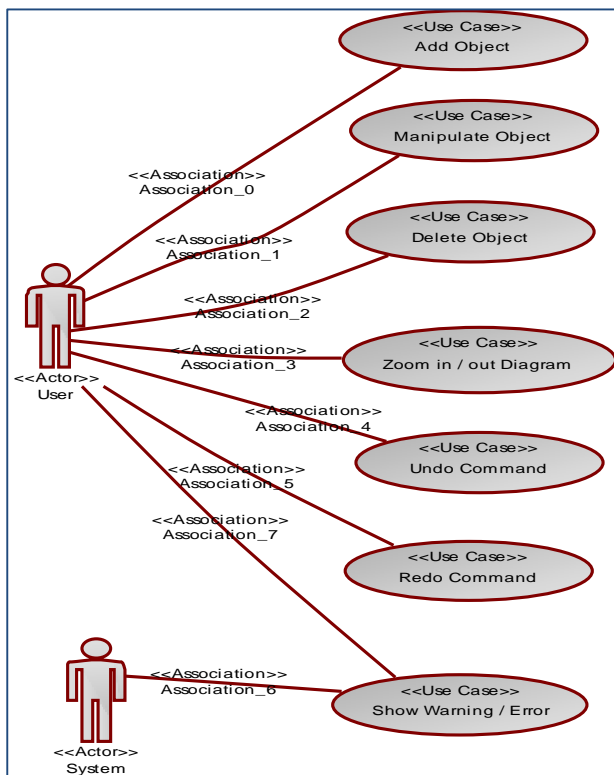
Inti dari proses konversi bolak-balik yang dilakukan adalah pada *Layer Data CLM* yang kemudian mengembalikan informasi ke *Layer Model* dan *Layer View*.



Gambar 2. Arsitektur Baseline Editor

3. Pemodelan Sistem

(1) Diagram *Usecase* Editor ditunjukkan pada Gambar 3 berikut ini.



Gambar 3. Diagram usecase Editor Model Data

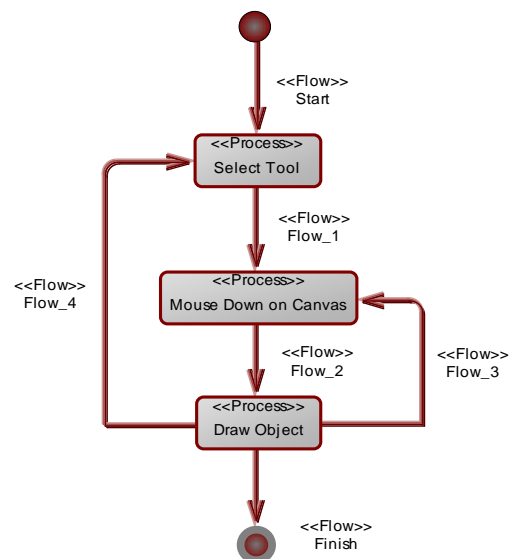
Editor ini memiliki use case sebagai berikut:

- *Add Object*, menambahkan obyek pada editor canvas,
- *Manipulate Object*, memanipulasi dan melakukan editing pada obyek yang ada pada editor canvas,

- *Delete Object*, menghapus obyek dari editor canvas,
- *Zoom in / out Diagram*, memperbesar tampilan diagram atau memperkecilnya.
- *Undo Command*, mengembalikan editor pada state sebelumnya.
- *Redo Command*, melanjutkan editor pada state berikutnya.
- *Show warning / Error*, menampilkan pesan *error* dan *warning*.

(2) Pemodelan aktivitas, ditunjukkan pada beberapa gambar berikut ini (Gambar 4 - 7).

Add Object: Menambahkan object pada diagram canvas, memiliki alur sebagai berikut.

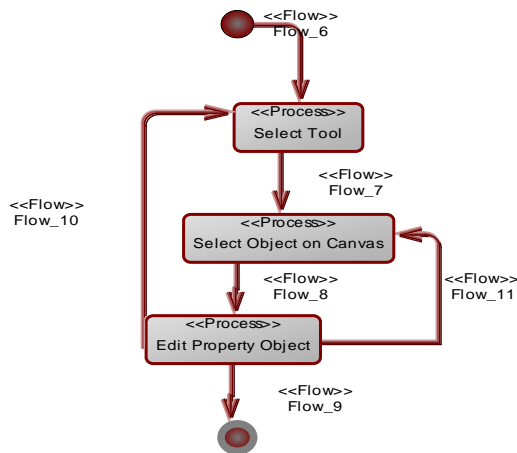


Gambar 4. Activity Diagram Add Object

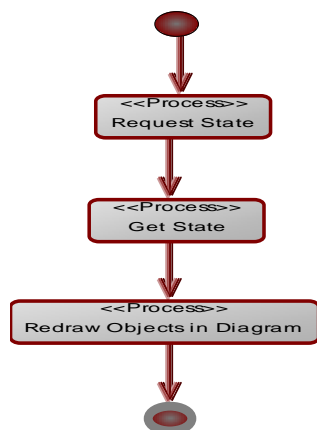
Obyek yang digambar sesuai dengan tool yang dipilih, contoh: Tool Entity akan menggambar Entity.

Manipulate Object: Memanipulasi object, adalah melakukan perubahan pada property object. Misalkan perubahan name, code, cardinality pada object. Sebelum melakukan perubahan pada suatu obyek, langkah yang harus dilakukan adalah melakukan seleksi pada obyek tersebut. (Gambar 5)

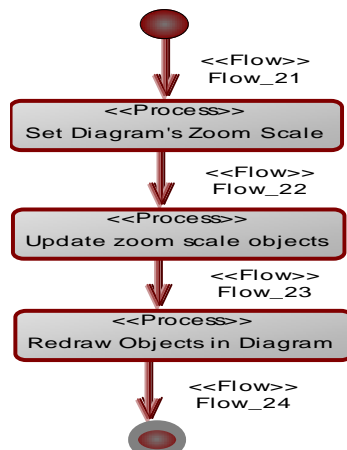
Undo dan Redo Command: Intisari dari undo dan redo command adalah mendapatkan state-state yang telah dilakukan dan mengaktifkan kembali state tersebut. (Gambar 6)



Gambar 5. Activity Diagram Manipulate Object



Gambar 6. Activity Diagram Undo Redo Command



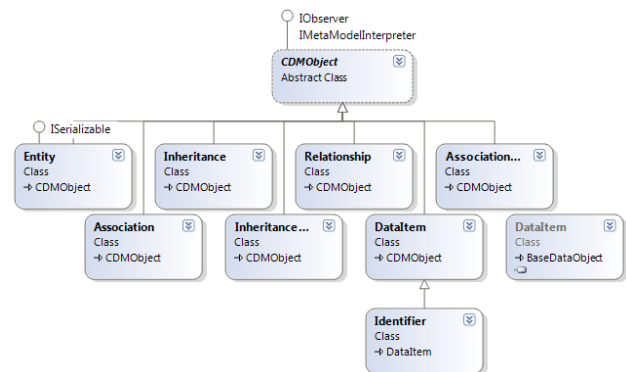
Gambar 7. Activity Diagram Zoom in / out Diagram

Zoom in / out Diagram: Zoom in dan Zoom Out pada saat melakukan desain sangat penting, untuk

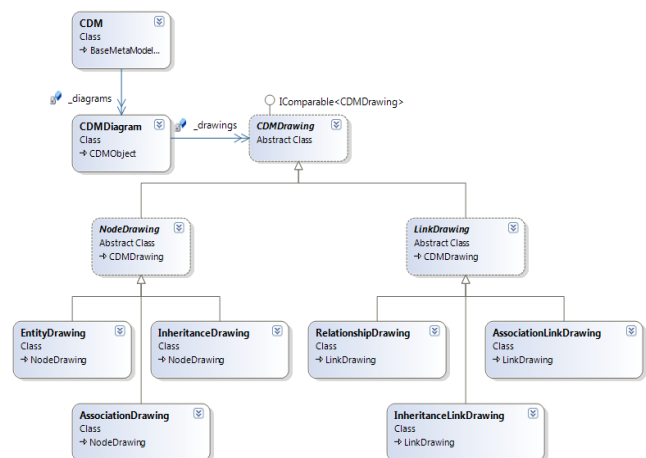
mengetahui secara detail informasi yang ada. Fitur ini juga membantu seorang desainer mengatur dan menempatkan obyek-obyek pada tempat yang diinginkan. Mekanisme zoom in / out adalah dengan merubah skala perbesaran pada diagram, kemudian perubahan itu diteruskan pada perubahan skala pada masing-masing obyek yang akan berpengaruh pada penggambarannya. (Gambar 7)

(3) Representasi dari Obyek-Obyek Pada CDM

Gambar-gambar berikut menunjukkan rancangan kelas dari elemen-elemen pada CDM.



Gambar 8. Rancangan Diagram Kelas pada CDM



Gambar 9. Diagram Class Drawing CDM

Kelas-kelas yang ada pada gambar di atas adalah representasi sederhana dari obyek-obyek yang ada pada Konseptual Data Model. Kelas-kelas ini berada pada Layer Model pada domain CDM.

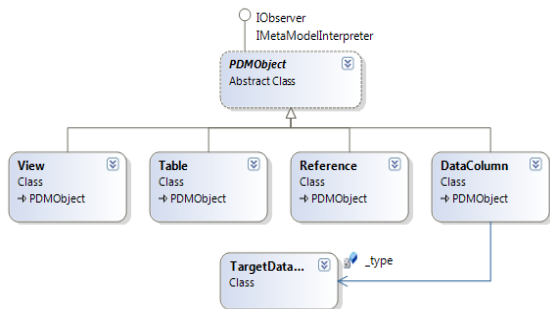
Pada Gambar 8 diatas, semua obyek diturunkan dari CDMObject. Entity, Association, Inheritance, Inheritance Link, Relationship, DataItem, AssociationLink, semuanya merupakan obyek-obyek pada CDM dan merupakan turunan dari CDMObject. Class-class pada Gambar 8, berasosiasi dengan class-

class pada Gambar 9 sebagai model pada drawingnya. Setiap Drawing memiliki class model di dalamnya. Contoh: class Entity Drawing memiliki field berupa class Entity di dalamnya.

Desain class Gambar 8 dan Gambar 9 menggunakan sebuah pola, yaitu pola strategi (strategy pattern). Pola strategi merupakan sebuah pola yang mendefinisikan sebuah family dari algoritma, dengan melakukan enkapsulasi pada masing-masing obyek dan menjadikan masing-masing obyek tersebut dapat diganti dengan mudah. Pola strategi ini merupakan sebuah algoritma yang terpisah dari client yang menggunakannya. Tujuannya adalah mendefinisikan kelas yang mengenkapsulasi algoritma line breaking yang berbeda-beda [1].

(4) Representasi dari Obyek-Obyek Pada PDM

Gambar-gambar berikut menunjukkan rancangan kelas dari elemen-elemen pada PDM.



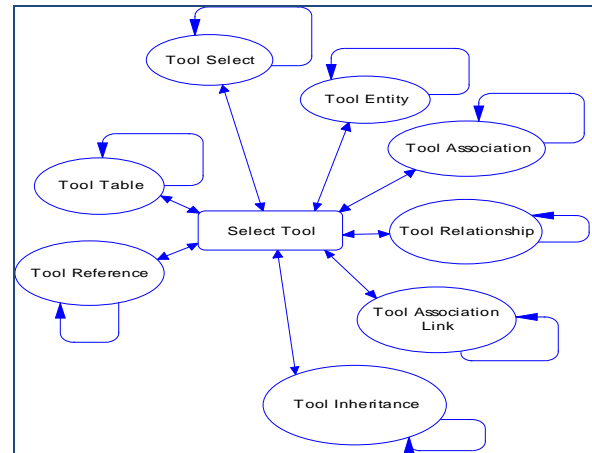
Gambar 10. Rancangan diagram class PDM

Pada Gambar 10 di atas seluruh obyek pada PDM diturunkan dari PDMObject. View, Table, Reference dan DataColumn, semuanya adalah obyek-obyek pada PDM dan semuanya merupakan turunan dari PDMObject. Rancangan obyek-obyek pada PDM, juga menggunakan pola Strategy sebagaimana pada CDM.

Semua class representasi object pada PDM dibungkus dengan class drawing masing-masing object. Class-class drawing pada PDM juga mengimplementasikan strategy pattern di dalamnya.

(5) Tool Yang Digunakan Dalam Editor

Untuk memenuhi kebutuhan fungsional sebuah editor model data yaitu kemampuan untuk menambah obyek diagram, manipulasi obyek diagram, dan penghapusan obyek diagram. Maka, untuk membedakan dibuat status tool. Diantara status tool yang ada adalah tool add untuk menambahkan setiap obyek pada diagram, dan tool select untuk melakukan seleksi pada obyek diagram.



Gambar 11. State Diagram Tools

Tool add meliputi penambahan setiap obyek riil pada masing masing diagram. Contoh, penambahan Entity, Association, Relation, dan Inheritance. Tool Select digunakan untuk melakukan seleksi pada obyek diagram, melakukan manipulasi obyek diagram, dan juga dapat digunakan untuk melakukan penghapusan obyek dari diagram.

Gambar 11 menjelaskan diagram state dari tool dan perpindahannya. Perpindahan state dari satu state ke state yang lain dengan melalui select tool. Pada state tool, tool selain tool select merupakan tool add. Tool Entity, tool association, tool relationship, dan selainnya, merupakan tool untuk menambahkan object pada diagram canvas.

(6) Perancangan Proses Manipulasi Pada Diagram Editor

Proses manipulasi obyek pada diagram dilakukan setelah dilakukan seleksi pada obyek. Manipulasi yang dilakukan adalah perubahan pada nama obyek dan properti-property yang dimiliki oleh obyek.

Ada beberapa metode untuk melakukan editing pada obyek dan propertinya, diantaranya dengan menggunakan form properti dan popup properti. Popup properti digunakan untuk melakukan pengeditan dan manipulasi dari properti visual, yaitu properti yang ditampilkan pada diagram. Ini lebih sederhana dari pada menampilkan form untuk melakukan manipulasi. Alur Manipulasi obyek dijelaskan pada Gambar 5.

- Mekanisme Show Pop up pada Diagram

Pop up property muncul ketika sebuah obyek pada diagram diseleksi, pada Gambar 5 dijelaskan pada saat activity Select Object On Canvas. Desain Pop up dibuat sederhana untuk melakukan editing pada properti-property object-object yang tampak pada diagram (visual).

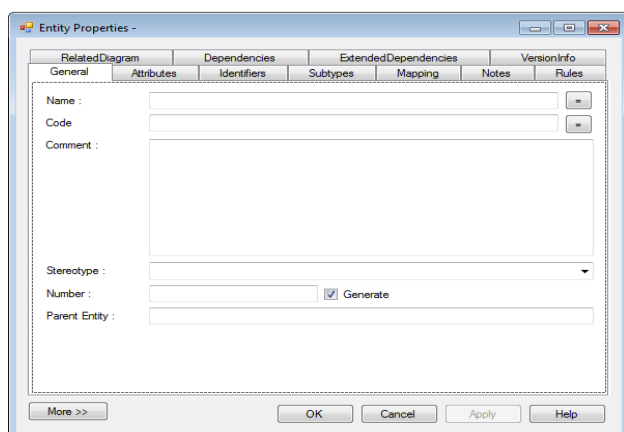


Gambar 12. Pop Up Entity

Setiap Popup memiliki pointer ke sebuah Object, yaitu object yang telah diseleksi. Jika telah selesai dilakukan pengeditan maka object pointer pada popup akan di update sesuai dengan perubahan yang dilakukan.

- Mekanisme Show Properties Form

Properties form digunakan sebagai media manipulasi object yang lebih detail dibandingkan dengan Pop up, yang hanya untuk editing properti-properti yang tampak pada diagram canvas. Direncanakan Pop up properties dibuat menggunakan User Control pada .NET framework [3]. Mekanisme show yang dilakukan hampir sama dengan mekanisme show Pop up. Untuk membedakan dari Popup, cara menampilkan Properties Form didahului dengan event double click. Contoh tampilan properties form pada Gambar 13.



Gambar 13. Properties Form Entity

4. Kesimpulan

Editor model data digunakan untuk memudahkan desainer basis data melakukan desain pada. Merupakan fungsional dari editor adalah dapat melakukan melakukan penambahan, manipulasi dan penghapusan obyek diagram. Fungsi-fungsi inilah yang harus ada pada sebuah editor model data. Fitur-fitur Pop up dan Properties Form digunakan untuk memudahkan pengguna melakukan editing dan manipulasi sebuah object. Demikian pula tambahan fitur undo dan redo command untuk mempermudah perubahan dan cancel perubahan pada diagram, dengan mengembalikannya ke state yang sebelumnya.

Fitur zoom in / out membantu seorang desainer mengetahui detail suatu object dan memudahkan pengaturan dan penempatan object-object pada diagram canvas. Sehingga, diagram yang dihasilkan bisa sangat mudah dimengerti dengan penyusunan yang dan penempatan masing-masing object dengan baik.

Secara Umum fitur-fitur yang direncanakan dimaksudkan untuk memudahkan user dalam melakukan desain sebuah model data dengan sebuah tool yang memanfaatkan dan memaksimalkan fungsi round trip engineer atau konversi bolak balik yang dilakukan secara realtime.

5. Rencana Pengembangan Selanjutnya

Rencana pengembangan selanjutnya dari pengembangan editor ini adalah:

- Implementasi Perangkat Lunak: Implementasi perangkat lunak dilakukan berdasarkan perencanaan dan perancangan sistem yang telah dibuat.
- Uji Coba Perangkat Lunak: Uji coba perangkat lunak dilakukan dengan pengujian fitur-fitur yang berkaitan dengan editor, dengan melibatkan sejumlah obyek model data (Black Box Testing).

Penghargaan

Penelitian ini terlaksana atas bantuan dana dari LPPM (Lembaga Penelitian dan Pengabdian kepada Masyarakat) Institut Teknologi Sepuluh Nopember Surabaya melalui dana PUM ITS 2011. Disamping itu, penelitian ini didukung oleh tim pengembang perangkat lunak CASE Tool "Nasty Steroid" Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya.

Referensi

- [1] Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Holland: Addison Wesley Longman, Inc., 1998.
- [2] Grinstein G, Georges. Database issues for data visualization: IEEE Visualization '93 workshop, San Jose, California, USA. 1993.
- [3] Troelsen A. *Pro C# 2008 and the .NET 3.5 Platform*. 4th Edition. New York: Apress; 2007.